

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-88863

(43)公開日 平成5年(1993)4月9日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 9/06

識別記号

4 3 0 E 8944-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 14 頁)

(21)出願番号 特願平3-249410

(22)出願日 平成3年(1991)9月27日

(71)出願人 000104674

キョーラク株式会社

京都府京都市上京区烏丸通中立売下ル龍前  
町598番地の1

(72)発明者 小川 敏

大阪府茨木市北春日丘4-8-3

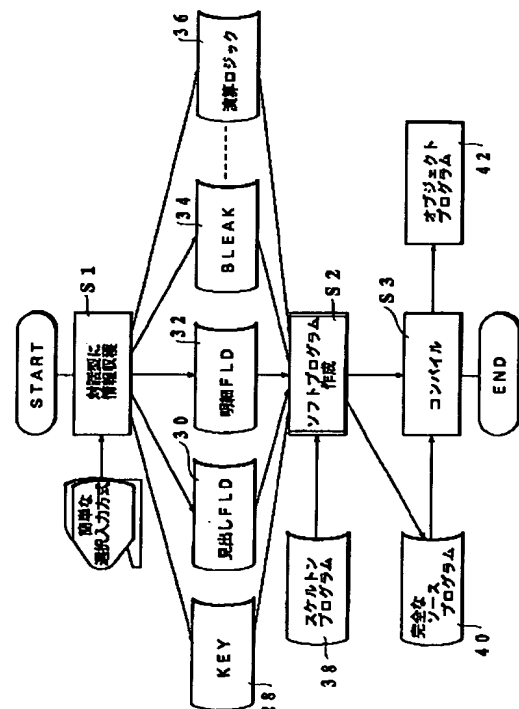
(74)代理人 弁理士 古谷 栄男

(54)【発明の名称】 プログラム開発支援システム

(57)【要約】

【目的】プログラム開発を容易にするプログラム開発支援システムを提供することを目的とする。

【構成】まず、プログラムの共通部分を抽出し、独自部分は未完成であるスケルトンプログラムを用意しておく。対話型の質問入力により、当該プログラムに固有の独自情報を得る(ステップS<sub>1</sub>)。この独自情報に基づいて、独自情報のファイル28, 30, 32, 34, 36を生成する。次に、これらの独自情報ファイル28, 30, 32, 34, 36に基づいて、スケルトンプログラムの未完成部分を完成させる(ステップS<sub>2</sub>)。次に、これをコンパイルしてオブジェクトプログラムを得る(ステップS<sub>3</sub>)。



**【特許請求の範囲】**

**【請求項1】** 目的とするプログラムを共通部分と独自部分に分け、  
共通部分によって構成したスケルトンプログラムを用意しておくとともに、  
独自部分に関する独自情報を収集するための独自情報収集手段、  
独自情報収集手段によって得られた独自情報に基づいて、スケルトンプログラムに独自部分を付加して、目的とするプログラムを完成させるプログラム完成手段、を備えたことを特徴とするプログラム開発支援システム。

**【発明の詳細な説明】****【0001】**

**【産業上の利用分野】** この発明は、プログラム開発支援システムに関するものであり、特にそのプログラム開発の容易化に関するものである。

**【0002】**

**【従来の技術】** コンピュータは、目的に応じてプログラムを変えることにより、種々の目的に対応できるという汎用性を有している。すなわち、ハードウェアを変更・追加することなく、ソフトウェアであるプログラムを変更するだけで、異なる機能を有する装置として使用することができる。

**【0003】**

**【発明が解決しようとする課題】** しかしながら、一般に、コンピュータプログラムの開発には、多くの時間が必要である。このため、必要な時までにはコンピュータプログラムが完成しなかったり、膨大な費用がかかったりするという問題点があった。

**【0004】** この発明は、上記のような問題点を解決して、プログラム開発を容易にするプログラム開発支援システムを提供することを目的とする。

**【0005】**

**【課題を解決するための手段】** この発明に係るプログラム開発支援システムは、目的とするプログラムを共通部分と独自部分に分け、共通部分によって構成したスケルトンプログラムを用意しておくとともに、独自部分に関する独自情報を収集するための独自情報収集手段、独自情報収集手段によって得られた独自情報に基づいて、スケルトンプログラムに独自部分を付加して、目的とするプログラムを完成させるプログラム完成手段、を備えたことを特徴としている。

**【0006】**

**【作用】** プログラム完成手段は、独自情報収集手段によって得られた独自情報に基づき、予め用意されたスケルトンプログラムの独自部分を完成させる。したがって、独自情報を与えるだけで、プログラムを完成することができる。

**【0007】**

**【実施例】** 図2に、この発明のプログラム開発支援システムにおいて用いたハードウェアの一例を示す。バスライン14を介して、CPU4、ROM2、RAM6が接続されている。また、バスライン14には、ディスプレイ(CRT)8、キーボード10、ハードディスク12も接続されている。ハードディスク12には、独自部分のプログラム内容が\*印によって置き換えられたスケルトンプログラムが記憶されている。ここで、種々の目的のプログラムにおいても基本的に異なることのないステップを共通部分とし、種々の目的のプログラムによって異なる固有のステップを独自部分とする。すなわち、スケルトンプログラムは、共通部分を抽出して構成されたプログラムであるといえる。

**【0008】** 図3および図4に、RPGによるスケルトンプログラム（一部分）の一例を示す。図3において、16, 18が未完成部分である。また、1～5桁目には、処理コード20, 22が記述されている。本来、RPGにおいては、1～5桁目には、注釈が記述される。このシステムにおいては、この注釈を記述する部分を利用して、処理コードを記述するようにしている。

**【0009】** 図4Aは、スケルトンプログラムの他の部分の一例であり、図4Bはこれによって生成されたプログラムである。行 $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $\delta$ が未完成部分である。図3と同様に、1～5桁目には、処理コード50, 52, 54, 56が記述されている。

**【0010】** ここで、処理コードとは、プログラムのステップまたはロジックを作成（変更）するためのコードである。これに対し、RPGにおいて用いられている、仕様書コードは、各ステップまたはロジックがどのような意味を持つのかを見分けるためのコードである（図3の21参照）。したがって、処理コードは、従来より知られている仕様書コードとは、異なるものである。

**【0011】** 図5に、処理コードとこれに対応する処理内容とを示す。処理コードは、処理タイプ別に変更処理のもの（1桁目がMのもの）と、作成処理のもの（1桁目がGのもの）とに大きく分けられる。2～3桁目は、詳細な処理内容を表わしており、4～5桁目は、処理対象となる桁数を表わしている。例えば、図3の処理コード20は、表示（もしくは出力）ファイル名を、33桁目からセットすることを表わしている。

**【0012】** 以上のように、スケルトンプログラムは、独自部分に\*印が記述された未完成ステップと、これに対する処理内容を記述した処理コードを有している。

**【0013】** 次に、このシステムを用いて、プログラム開発を行う際のフローチャートを図1に示す。まず、ステップS<sub>1</sub>において、CPU4はディスプレイ8に質問事項を表示し、プログラム作成者からの回答入力を得る。

**【0014】** このように、独自情報収集手段は、質問事項に回答する対話型作業により独自情報を収集する。独

自情報における見出しフィールド、明細フィールド、キーフィールド、見出し（又は明細）フィールド入力により参照する参照ファイル、参照ファイルに連鎖するキーフィールド、参照するフィールドなどのファイルおよびフィールドは、質問事項のなかの画面上に自動的に表示された選択項目を見ながら指定する方法が採用される。

【0015】独自情報を収集するため、予め設定された手順で画面に表わされる質問事項の主な項目名を以下に示す。

【0016】(1)アクセスするファイル名の入力

(2)キーフィールドの選択

(3)見出しフィールドの選択

(4)明細フィールドの選択（図6B参照）

(5)参照フィールドの選択（図7A参照）

(6)連鎖するフィールドの選択（図7B参照）

(7)参照するフィールドの選択

(8)表示・出力順序の入力（図8A参照）

(9)表示・出力の追加情報の入力（図8B参照）

なお、アクセスするファイル名とは、更新業務・照会業務また帳票作成業務において、予めその業務に必要なデータが書き込まれた任意のキー付きファイルである。キーフィールドとは、ファイルをアクセスするためのフィールドでファイルと呼び出し、更新、照会、出力等を行うものである。見出しフィールドとは、画面（あるいは帳票）の上部に見出しとして1回だけ表示するフィールド（図9の22の部分）であり、明細フィールドとは、画面（あるいは帳票）の下部に何度も繰り返して表示されるフィールド（図9の24の部分）である。

【0017】すなわち、アクセスするファイル名を入力した後は、キー付きファイルによって各フィールド、ファイルの入力を画面上の選択項目の指定により簡単に行えるという効果を有する。

【0018】例えば、図6Aに示すような表示をディスプレイ8上で行う。プログラム作成者は、これを見て、処理対象となるファイル名20等を入力する。入力されたデータは、RAM6に記憶される。

【0019】1つの画面に対する回答入力が終わると、CPU4は次に、図6Bに示す質問画面をディスプレイ8に表示して入力を促す。図6Bにおいては、ファイルのフィールドの内、明細フィールドとして選択するものにX印を入力させるようにしている。

【0020】次に、図7Aおよび図7Bの画面によりファイル間の結合定義を行う。以下同様にして、図8A、図8Bにより、見出しフィールド、明細フィールドの表示順序等を入力する。

【0021】以上のようにして対話形式により入力された独自情報は、キーファイル28、見出しファイル30、明細ファイル32、ブレイクファイル34、演算ロジックファイル36として、RAM6に記憶される。

【0022】次に、CPU4は、上記の独自情報による

各ファイルに基づいて、未完成のスケルトンプログラム38を完成させる（ステップS<sub>2</sub>）。この部分のフローチャートの詳細を、図10に示す。まず、ステップS<sub>10</sub>において、スケルトンプログラム38を1ステップずつ読み込む。読み込んだスケルトンプログラム38の1桁目が「G」であるか、「M」であるか、それ以外の文字であるかを判断する（ステップS<sub>11</sub>）。「G」であれば、図5の処理方法にしたがって、ステップを生成する（ステップS<sub>12</sub>）。

10 【0023】ステップが生成される状態を、図4を用いて説明する。まず、行αの2桁目から3桁目が「BL」であるので、CPU4は、明細ファイルの項目にデータが入っているか否かのチェックを行うステップを生成する処理コードであることを知る（図5参照）。次に、CPU4は、明細ファイルの項目名、属性を読み込む。読み込んだ属性に基づき、明細ファイルの項目のうち、入力に使用する項目を選択する。ここでは、URIHIN, URISRY, URITAN, URIKINの各項目が入力に使用されるものであるとする。

20 【0024】次に、CPU4は、項目名「URIHIN」を18桁目から書き込む（図4Bの60参照）。さらに、28桁目からRPGにおける比較命令「COMP」を書き込む（図4Bの62参照）。どのような命令を書き込むかは、処理コードにより定められる。次に、33桁目からデータの型を書き込む。このデータの型は、読み込んだ明細ファイルの属性から判断する。以上のようにして、1行のステップが生成される。

30 【0025】以下同様にして、URISRY, URITAN, URIKINについてもステップを生成する。このようにして、4行のステップが完成する。

【0026】以下上記と同様にして、処理コード52, 54, 56に基づいて、図4Bに示すようにステップが生成される。

【0027】図10のステップS<sub>12</sub>において、「M」であれば、図5の処理方法にしたがって、ステップを完成させる。例えば、図3において、処理コード20が記述されたステップを読み込んだ場合には、キーファイル28から読み出されたファイル名を、未完成部分16の\*印と置き換えて、ステップを完成させる。その後、ステップS<sub>13</sub>を実行する。なお、1桁目が「G」「M」以外の文字である場合には、何も処理を施さず、ステップS<sub>15</sub>を実行する。

【0028】ステップS<sub>12</sub>においては、1桁目が「G」「M」であった場合には、1～5桁目を消去して、ハードディスク12への書き出しを行う。「G」「M」以外の文字であった場合には、1～5桁目を消去することなくハードディスク12への書き出しを行う。

【0029】以上のようにして、1つのステップについての処理が終了すると、ステップS<sub>16</sub>に戻って、同様にして、次のステップの処理を行う。これを、スケルトン

プログラム38の最後のステップまで繰り返し、終了する（ステップS<sub>11</sub>）。

【0030】上記の結果、完成したソースプログラム40がハードディスク12内に生成される。次に、このソースプログラム40をコンパイルする（図1のステップS<sub>3</sub>）。これにより、オブジェクトプログラム42が得られる。

【0031】一般的に、従来のスケルトンプログラムの開発方法であれば、入力エディターの画面に向けた状態で、多様なフィールド名を調べる作業、多種の演算命令、演算項目、結果項目、ファイル名、標識などからなるロジックを作成する作業、また、そのロジックを入力エディターの画面上の桁数、順序を意識しながら入力する作業を必要とすることから、多くの開発ステップ（手順）と開発時間が必要である。また、プログラムの開発作業も特定のプログラム言語およびアクセスするデータのファイル内容を理解した作業であることが必要である。

【0032】それに対し、実施例にて示したスケルトンプログラムの開発方法にあっては、スケルトンプログラム（プログラム共通部分）の中に独自部分を挿入、置換するための暗号である処理コードを入れておき、画面からの独自情報を、処理コードを使ってプログラムコード化し、スケルトンプログラムの部分的な置き換え、書き出し、消去を自動的にを行い、目的とするプログラムを完成する手段をとっている。これにより、従来の多くの開発ステップと開発時間を飛躍的に短縮することができる。また、プログラムの開発作業も従来のようなプログラム言語を習得した専門家である必要はない。すなわち、予め設定された手順で表われる画面上の簡単な質問にしたがって独自情報を入力する方法としている。よって、一般の事務担当者、経理担当者であってもソースプログラムを作成することができる。さらに、特定の処理コードを活用し、且つ、予め設定された手順で表わされる画面上の質問を選択するようにしている。したがって、スケルトンのパターンを増やすことによって、あるいは処理コードを増やすことによって、あるいは処理コードと対応する独自情報ファイルの種類を増やすことによって、多数の異なるプログラムを簡単に作成できる。さらに、アクセスするデータファイルの変更、また出力フォーマットの修正された他の類似するプログラムであっても、専門的な演算命令を入力エディターにて作成することなく、簡単に、変更また修正された他の類似する \*

\* プログラムを容易に作成することができる。

【0033】以上のように、この実施例では、対話型の質問に回答するだけで、プログラムを完成させることができる。

【0034】なお、上記実施例においては、RPGについて説明したが、C、コボル等の他の言語についても適用可能である。さらに、コンパイルを行う必要のない言語についても適用することが可能である。

【0035】

10 【発明の効果】この発明に係るプログラム開発支援システムにおいては、予め、スケルトンプログラムを用意しておき、入力された独自情報に基づいてプログラムを完成するようにしている。したがって、独自情報を与えるだけで、プログラムを完成することができ、迅速にプログラム開発を行うことができる。

【0036】すなわち、この発明によれば、プログラム開発を容易にするプログラム開発支援システムを提供することができる。

【図面の簡単な説明】

20 【図1】この発明の一実施例によるプログラム開発支援システムのフローチャートである。

【図2】この発明の一実施例によるプログラム開発支援システムのハードウェア構成を示す図である。

【図3】スケルトンプログラムの一例を示す図である。

【図4】スケルトンプログラムの一例を示す図である。

【図5】処理コードの内容を示す図である。

【図6】ディスプレイ8上に表示される質問画面を示す図である。

30 【図7】ディスプレイ8上に表示される質問画面を示す図である。

【図8】ディスプレイ8上に表示される質問画面を示す図である。

【図9】データ表示の一例を示す図である。

【図10】ソースプログラム作成部分の詳細を示すフローチャートである。

【符号の説明】

2・・・ROM

4・・・CPU

6・・・RAM

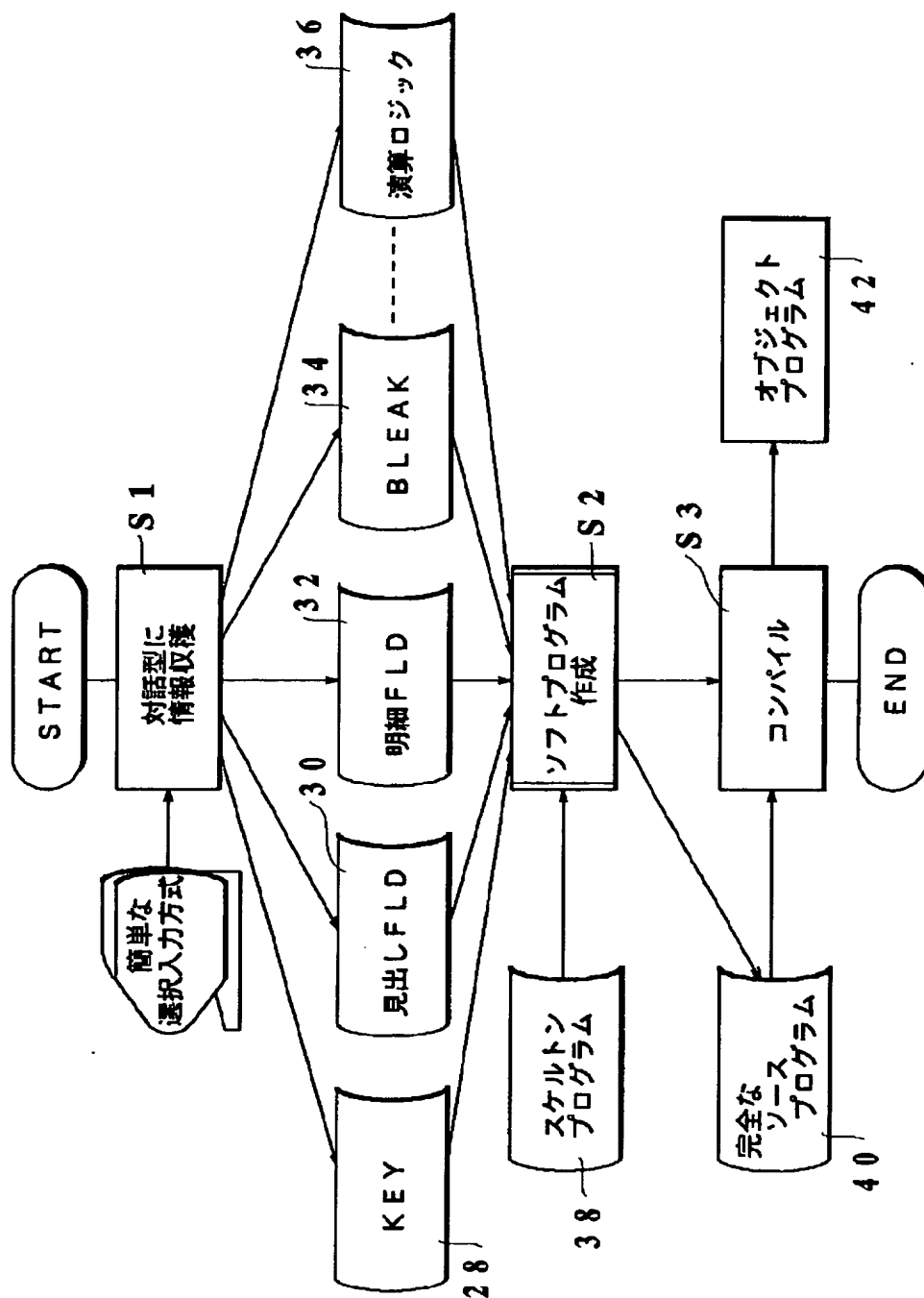
40 8・・・ディスプレイ

10・・・キーボード

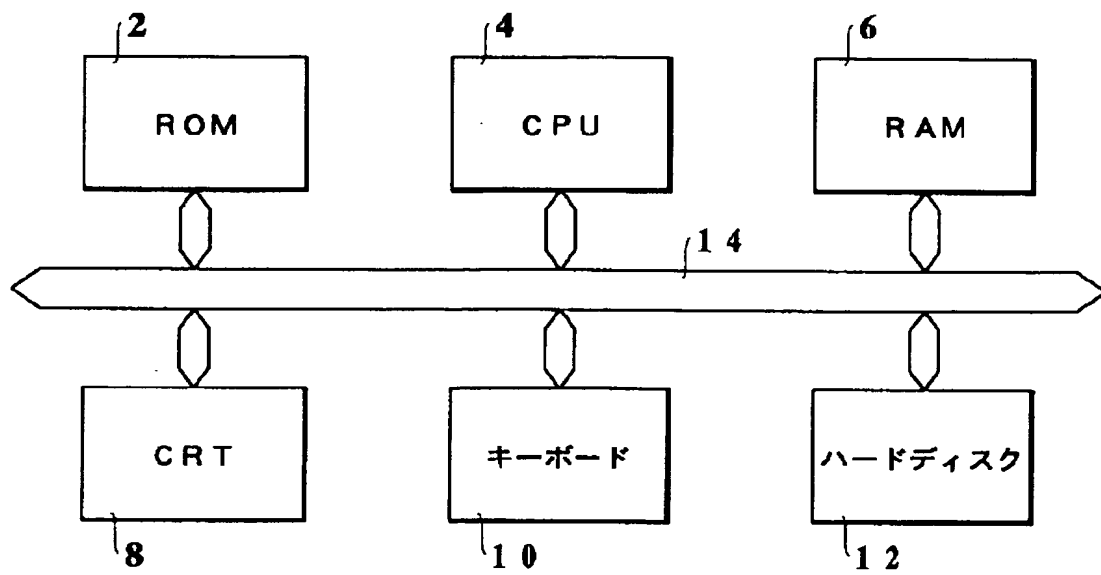
12・・・ハードディスク

38・・・スケルトンプログラム

【図1】



【図2】



桁	1	2	3	4	5	6	7	8	9	10	.....
8100							C	*	INPUT	CHECK	
8200							C				
8300							C				
8400							C		N20	IO	
8500							C				
8600							C		20	FULKEY	
8700							C				
8800							C		22		
8900							C				
9000							C				
9100							C				
9200							C				
9300							C				
9400							C		N20		

【図4】

**A**

	50	18	28	33	43
α	C	GBD	SETON		30
	C	C*	*IN30	IFEQ '0'	1F
β	C	C*	DB-FIELD TO WORK-STATION FIELD		
	C	GPCDDC*			
	C	99	MOVE ERR 4	ERRMSG	
γ	C	GLOADC*	UPDATSFLREC		
	C	54	ELSE		
δ	C	GMB	UPDATSFLREC		
	C	56	END		1F

**B**

	C	30	SETON		30
αから 生成	C	URIKIN	COMP *BLANK		30
	C	URIKIN	COMP *ZERO	64	30
	C	URIKIN	COMP *ZERO		30
	C	60	URIKIN 62	COMP *ZERO	30
	C	*IN30	IFEQ '0'		1F
	C	DB-FIELD TO WORK-STATION FIELD			
βから 生成	C	MOVE *BLANK	HINMIO		
	C	RKEY60	CHAINHINLE1		60
γから 生成	C	SETON			99
	C	MULT	URISRY	URIKIN	
	C	MOVE ERR 4	ERRMSO		
	C	UPDATSFLREC			
δから 生成	C	ADD	URIKIN	WST001	
	C	ELSE			
	C	MOVE *BLANK	URIHIN		
	C	MOVE *BLANK	HINMIO		
	C	MOVE *ZERO	URISRY		
	C	MOVE *ZERO	URITAN		
	C	UPDATSFLREC			
	C	END			1F



【図5】

TYP	MOD	POS	処理方法
M (変更)	DS LN DB BZ	07 33 07 33	画面ファイル名を7桁目からセット 明細行数を33桁目からセット 表示(出力)ファイル名を7桁目からセット 画面がブランク・ゼロのチェックの為に*BLANK又は*ZERO WO33桁目からセット
G (作成)	KY  KL IZ MV PC BL TO MB	FL LV HV DB ST ED	<p>キーのデータストラクチャ作成 (ポインターセット用) (LOVAL用) (HIVAL用) (ファイルとの比較用) (画面の一番上用) (画面の一番下用)</p> <p>他...</p> <p>各ファイルのキーストロジック作成 表示(出力)内容初期設定 表示(出力)内容セット 明細演算ロジック作成 明細が入っているかどうかブランクチェック 合計項目に加算する 明細全項目にブランクセット</p>

【図6】

**A**

入力更新プログラム 自動作成 (GNTENT)

選択項目を して、実行キーを押してください。

更新するファイル名	:	:	:	:	:	UR1LF1	NAME
参照するファイル名	:	:	:	:	:	TOKLF1	NAME
						H1NLF1	
ファイルを含むライブラリー名	:	:	:	:	:	JSXXX	NAME
作成するプログラムメンバー名	:	:	:	:	:	ENTO2	NAME
プログラムソース名	:	:	:	:	:	QRPGSRC	NAME
プログラムライブラリー名	:	:	:	:	:	JSXXX	NAME
作成する画面メンバー名	:	:	:	:	:	ENTO2FM	NAME
画面ソース名	:	:	:	:	:	QDDSSRC	NAME
画面ライブラリー名	:	:	:	:	:	JSXXX	NAME
作成後SEUで修正する	:	:	:	:	:	N	Y, N
画面プログラムのコンパイルを行なう	:	:	:	:	:	Y	Y, N
コンパイル結果のライブラリー名	:	:	:	:	:	JSXXX	NAME

20

終り

F 3 = 終了 F 4 = プロンプト F 5 = 再表示 F 1 2 = 取消し F 1 3 = この画面の  
F 2 4 = キーの続き 使用法

**B**

入力更新プログラム 自動作成  
売上伝票入力

8. 明細フィールドを選択しなさい(Xを入れる)

X	DBフィールド	長さ	属性	小数	見出し
	URIDAT	6	S	00	伝票日付
	URIDEN	7	A		伝票No
	URITOX	7	A		得意先コード
X	URIHIN	8	A		品目コード
X	URISRY	5	P	00	数量
X	URIYAN	7	P	02	単価
X	URIKIN	7	P	00	売上金額

22

F 3 : 最初の画面へ

【図7】

**A**

入力更新プログラム 自動作成  
売上伝票入力

4. 見出しフィールド入力により参照するファイルがあれば  
 選択しなさい(Xを入れる)

X	参照ファイル
(X)	TOKLF1
	HINLF1

F3 : 最初の画面へ

**B**

入力更新プログラム 自動作成  
売上伝票入力

5. 参照ファイルに連鎖するキーフィールドを  
 選択しなさい(Xを入れる)

参照ファイル	TOKLF1	
キーフィールド	TOKCD	得意先コード

X	DBフィールド	見出し
(X)	URIDEN	伝票No
	URITOK	得意先コード

F3 : 最初の画面へ

【図8】

A

入力更新プログラム 自動作成  
売上伝票入力

## 14. 明細フィールドの画面上左からの表示順序を定義する

(左: 見出 右: フィールド)

フィールド	順序	長さ	属性	小数	見出し	表示属性	編集	合計
URIHIN	1. 0	8	A		品目コード	R1 CS	—	—
URISRY	2. 0	5	P	00	数量	R1 CS	J	—
URITAN	3. 0	7	P	02	単価	R1 CS	J	—
URIKIN	4. 0	7	P	00	売上金額	R1 CS	J	X
NINH10	1. 5	22	O		品目名	R1 CS	—	—

F3: 最初の画面へ

F0: 表示属性・編集の説明

B

入力更新プログラム 自動作成  
売上伝票入力

## 14 明細フィールドの追加情報を入力する

表示位置、追加キーワードを指定しなさい (下段左: 見出 右: フィールド)

フィールド名	長さ	属性	小数	見出し	見出行	桁	フィールド	行	桁
URIHIN	8	A		品目コード	16	14		17	16
DSPATR (R1)							DSPATR (CS)		
HINMIO	22	O		品目名 (漢)	16	13		17	19
DSPATR (R1)									
URISRY	5	P	00	数量	16	14		17	14
DSPATR (R1)							DSPATR (CS)	EDTCDE (J)	
URITAN	7	P	02	単価	16	15		17	15
DSPATR (R1)							DSPATR (CS)	EDTCDE (J)	
URIKIN	7	P	00	売上金額	16	17		17	17
DSPATR (R1)							DSPATR (CS)	EDTCDE (J)	

明細の行間隔を指定しなさい (0-3) ..... 0

F3: 最初の画面へ

F5: イメージの表示

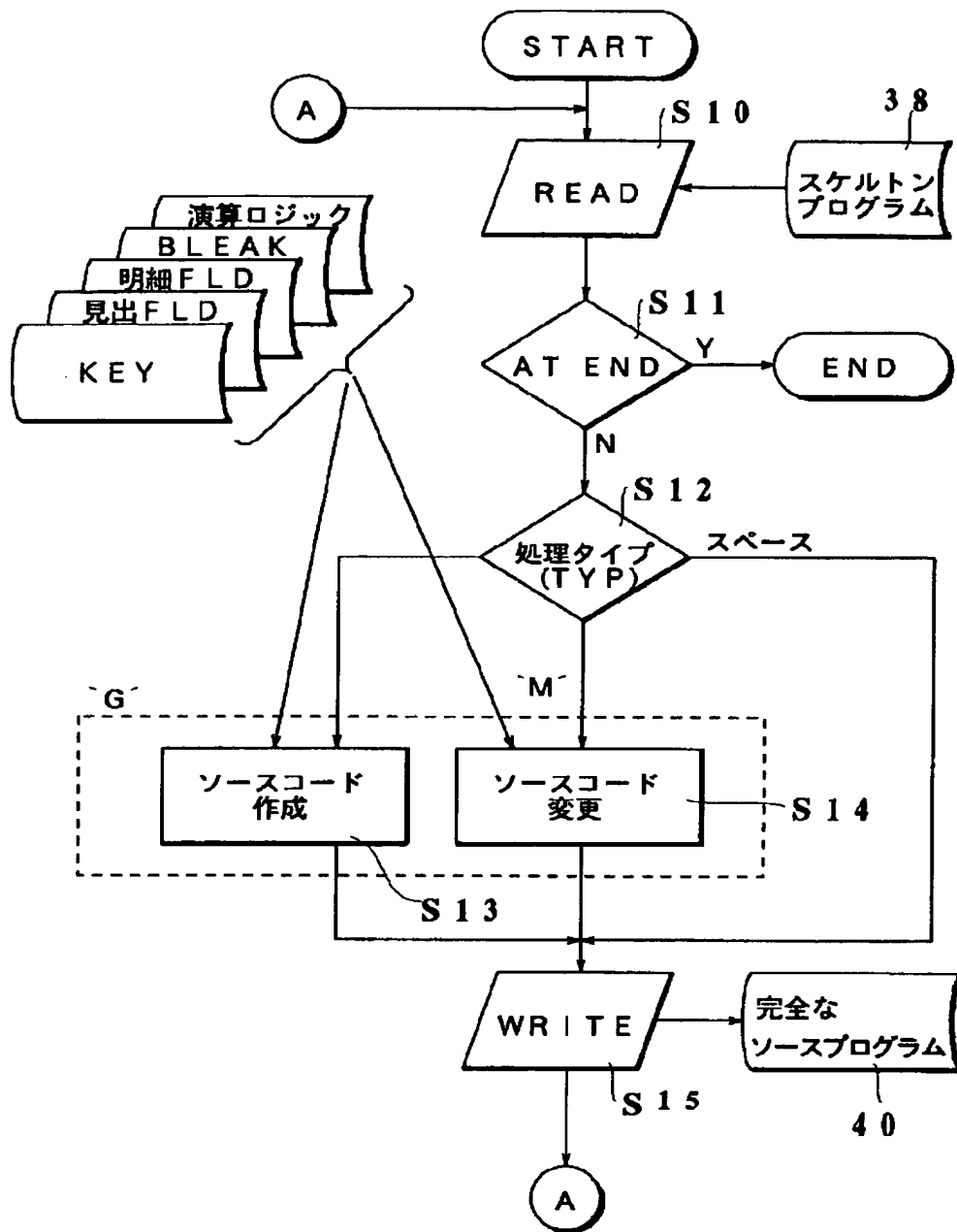
F9: 開始位置自動指定

F12: 前の画面へ

【図9】

ENT02		売上伝票入力		99/99/99	
22	伝票日付	910115			
	伝票No	3251			
	得意先コード	438			
	得意先名(漢)	キョーラク機			
24	品目コード	品目名	数量	単価	売上金額
	312	開発支援ツール	2	6,000,000	12,000,000
	308	導入費用	1	500,000	500,000
	105	交換プログラム	3	100,000	300,000
					12,800,000

【図10】



This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

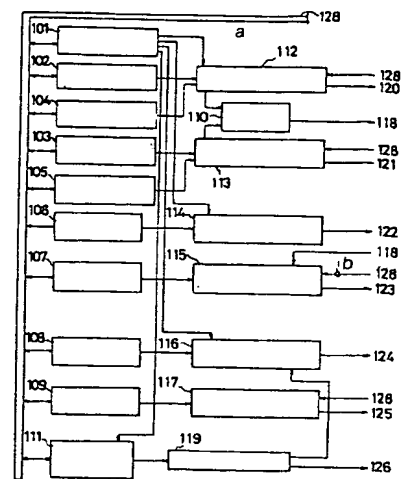
Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

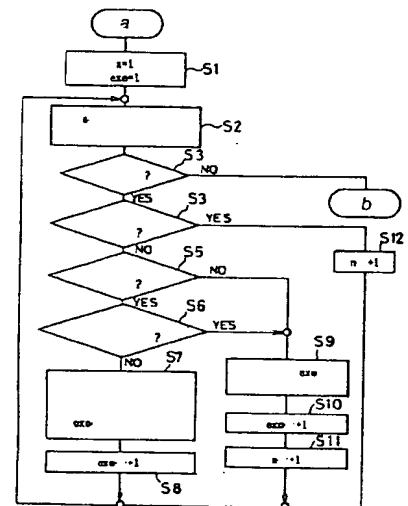
**CONSTITUTION:** This device has storing parts 101-103 to store the information such as a hardware block diagram and a port map table, storing parts 104-109 to store the standard software coding processing procedure such as the input output, logic, initialization, interruption and diagnosis, preparing parts 112-117 to refer to the storing parts 104-109 in accordance with these coding processing procedures and prepare the software part code, and means 111, 119 and 126 to synthesize these part codes and prepare the completed software code.



110: input output correspondence table preparing part, a:  
input output user interface part, b: solid data input

(51) Int. Cl.<sup>5</sup>. G06F9/06

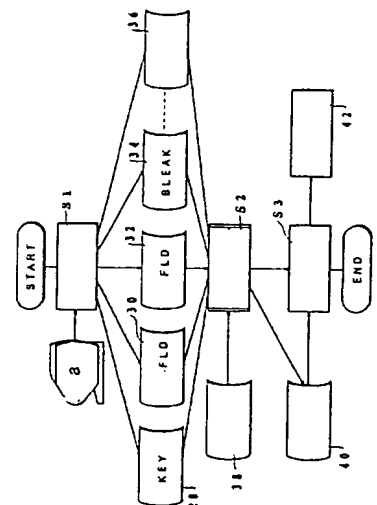
**CONSTITUTION:** By a control loop drawing device, control loop parts selected out of the control loop parts stored in a parts library are called and displayed, and the flow of data between the control loop parts called and displayed from the parts library is designated and displayed. The information concerning the control loop chart drawn and displayed by the control loop drawing device is registered into a drawing information file. From the information concerning the control loop chart registered in the drawing information file, the execution sequence between the control loop parts is automatically assigned by an execution sequence automatic assigning device, compiled to an instruction code in accordance with the execution sequence assigned automatically by a compiling device, and the control program is prepared.



a: start, b: end, S2: remove n-th parts information, S3: parts provided?, S3: the execution number provided?, S5: upstream data provided? S6: the execution number of upstream parts provided?, S7: Assign the execution number exe to the most upstream parts for which the execution number is not assigned., S9: assign the execution number exe.

(51) Int. Cl.<sup>5</sup>. G06F9/06

**CONSTITUTION:** First of all, the common part of the program is extracted, and a skelton program having an incomplete peculiar part is prepared. Peculiar information intrinsic for the relevant program is obtained by interactively inputting questions (S1). Next, files 28, 30, 32, 34 and 36 of the peculiar information are generated based on this peculiar information. Afterwards, the incomplete part of the skelton program is completed based on these peculiar information files 28, 30, 32, 34 and 36 (S2). Then, an object program is obtained by compiling this completed part (S3).



30: index FLD, 32: detailed FLD, 36: arithmetic logic.  
38: skelton program, 40: complete source program, 42:  
object program, (a): simple selective input system